Review of bioinformatics algorithms used in phylogenetic analysis and MSA

Aleix Giménez Grau*
ETSETB, Polytechnic University of Catalonia.

(Dated: May 28, 2014)

The problem of constructing phylogenetic trees is addressed from the point of view of bioinformatics, emphasizing their relation with multiple sequence alignments and presenting methods to solve both problems. A C++ implementation of some algorithms was developed and applied to real nucleotide data downloaded from GenBank. By these means the phylogenetic tree of the Hominidae family is recovered with some minor errors.

I. INTRODUCTION

Biology experimented a revolution in the middle of the past century starting with the discovery of the structure of DNA in 1953 by Watson and Crick, and followed by great technical achievements, for instance the sequencing of the first complete genome in 1976. Nowadays extensive genetic information is freely available online from GenBank[1], and the main problem is how to treat data in order to obtain relevant results. Work has been done in this field in the last decades, resulting in the appearance of a new discipline, i.e. *bioinformatics*, that aims at using computation in order to extract information from molecular biological data.

The study of evolution has been an important field since the appearance of Darwin's theory in 1859. Although comparative anatomy has been the main tool of evolutionary biology during a long period bioinformatics has changed completely this methodology. Nowadays species are no longer classified in terms of their anatomic characteristics but in terms of their genetic material, which has resulted in the concept of *phylogenetic trees*, i.e. evolutionary trees build in terms of genetic similarity. However, comparing the whole genome of species is still not possible, so shorter fragments must be used. Mitochondrial DNA is widely used in phylogenetics because it is shorter and more stable than a complete genome, and it has been determined from many species, even extincted ones.

In this paper, the methods for building phylogenetic trees based on bioinformatics will be reviewed. In order to do so multiple sequence alignments (MSA) must be analyzed in detail and their link with phylogenetics must be stressed.

The structure of the manuscript is the following: in section II the concept of pairwise alignment is defined and generalized to include multiple sequences. In section III the main algorithms to find alignments and phylogenetic trees are presented, i.e. Needleman and Wunsch algorithm for pairwise alignment, progressive alignment for MSA, the Jukes-Cantor model for constructing distance matrices and UPGMA and neighbor joining for phylogenetic tree building. Moreover, the implementation in C++ of these methods is summarized. In section IV the results obtained applying the code to genetic data from the Hominidae family are shown and discussed. Finally in section V conclusions are outlined briefly.

II. ALIGNMENTS. GENERALITIES

The first problem that arises when dealing with genetic sequences of different species is how to compare them. It is known that the origin of genetic diversity are mutations, that can take the form of substitution, deletion or insertion of base pairs. An alignment is a way of arranging the sequences of nucleotides or amino acids to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences[2].

We are going to discuss only nucleotide sequences, that can be thought as vectors of characters *a*, *c*, *g*, *t*. Given two sequences that can be of different lengths, an alignment of them is an array which two rows are strings of the same length with gaps in some positions, with the conditions that one column cannot contain two gaps and when removing the gaps the original strings must be recovered. Given the sequences *aggtactatccagtca* and *atgctagattaatgcaaggca* a pairwise alignment that follows the previous conditions can be, for instance

$$\begin{array}{c} ag\_\_\_\_gtact\_atcca\_gtca \\ atgctag\_attaatgcaaggca \end{array} \tag{1}$$

Given two sequences many alignments fulfill the previous definition, so *scoring functions* are needed to define which is the best. A scoring function takes two arguments that can be characters representing nucleotides or gaps and it returns a real number. For DNA sequences it is customary to use the scoring function in table I: when both arguments are equal it is a *match* and the score is one; when the arguments are different it is a *mismatch* and the score is $-\mu$; finally if one of the arguments is a gap it is an *indel* and the score is $-\sigma$. Although in this work it is assumed the most simple situation with $\mu = \sigma = 1$, there exists more advanced techniques, for instance using distinct penalties for opening a gap or extending it.

* aleixgim@gmail.com

δ	a	c	g	t	-
a	1	-μ	-μ	-μ	-σ
c	-μ	1	-μ	-μ	-σ
g	-μ	-μ	1	-μ	-σ
t	-μ	-μ	-μ	1	-σ
-	-σ	-σ	-σ	-σ	$\times$

TABLE I. Example of the simplest scoring function used in nucleotide alignments. In this work it is assumed $\mu = \sigma = 1$.

When a scoring function is selected the total score of the alignment is defined as the sum of the score of all sites. Namely, if the alignment array is $2\times l$ and its rows are s'_{1j} and s'_{2j} the total score is

$$S_{pw} = \sum_{j=1}^{l} \delta(s'_{1j}, s'_{2j}). \tag{2}$$

A generalization of the previous score is needed if the alignment consists on N sequences arranged in the array s'_{ij} of the multiple sequence alignment. The sum of pairs (SP) score is defined as the sum of the score over all pairwise alignments

$$S_{sp} = \sum_{i=1}^{N} \sum_{r=i+1}^{N} \left(\sum_{j=1}^{l} \delta(s'_{ij}, s'_{rj}) \right). \tag{3}$$

The pairwise alignment problem consists on finding the optimal position of the gaps in order to maximize the score S_{pw}, while the multiple sequence alignment problem maximizes S_{SP}. Although the formulation of both problems is very similar, the methods to solve them are significantly different, as will be seen in section III A.

III. MATERIALS AND METHODS

The algorithms used in this work will be described in this section, starting with techniques for pairwise and multiple sequence alignment, and then presenting tree-building algorithms.

A. Alignment techniques

One approach to solve the pairwise alignment problem is the Needleman and Wunsch algorithm[3], that reformulates it in terms of a directed acyclic graph (DAG), and finding its longest path is equivalent to solve the original problem. A dynamic programming algorithm is applied to the DAG so the alignment can be determined with computational complexity $O(l^2)$. This cost is not prohibitive for most purposes with the current hardware of personal computers.

In principle the multiple sequence alignment problem could be solved with Needleman and Wunsch algorithm, but the computational complexity would be of $O(2^n l^n)$, which restrict the application of the exact algorithm to only few short sequences. The problem has been shown to be NP-complete[4], which roughly implies that an efficient algorithm to solve it can not be found. This has derived to the use of approximate or heuristic approaches to find MSA, that make a trade-off between accuracy and efficiency.

Progressive multiple alignment is one of the alogrithms that solves the previous problem approximately. The main idea is that in the alignment of species with close phylogenetic relation few gaps need to be introduced, so these alignments are "better" than the ones rendered with distant species. This suggests that a possible strategy to MSA could be to pairwise align first the close sequences and then progressively "align this alignments" starting from closer to more distant groups of species. Progressive alignment warranties that at least the alignment of close species is carried properly, but convergence to the optimal solution may not be reached.

A more formal description of the algorithm is the following: if the phylogenetic tree of the species is known the sequences can be progressively aligned following its topology starting from leaves and arriving to the root. The phylogenetic tree previously used is called *guiding tree*, and can be obtained from a distance matrix method (see section III B). Most times a pairwise alignment of two arrays of multiple sequences must be done, and this is accomplished using Needleman and Wunsch algorithm maximizing SP-score of them. Using this approach the complexity of the algorithm is approximately $O(2^n l^2)$ which is again a reasonable cost for most purposes.

B. Distance-based tree-building methods

In the previous section the relation between phylogenetic trees and multiple sequence alignments was glimpsed when a guide tree was used to find an approximate MSA, but this link is tighter. A multiple alignment somehow displays genetic information in such a way that the evolution process that resulted in the studied sequences can be determined easily. In fact, if two rows of a MSA are almost identical they correspond to close species but if there are a lot of mismatches then they correspond to distant species.

When discussing the alignment of sequences scoring was used as a measure of the similarity between them. However, in phylogenetics it is more useful the use of distances, being the simplest one the *p-distance*, which is defined as the rate between the total number of mismatches and indels and the length of the alignment:

$$p = \frac{\sum_j p(s'_{1j}, s'_{2j})}{l}, \quad p(a,b) = \begin{cases} 1 & \text{if } a \neq b \\ 0 & \text{if } a = b \end{cases} \tag{4}$$

where a and b can be characters or gaps. The p-distance is purely mathematical and do not make any biological assumption. More advanced distances use a model of evolution of DNA, being the simplest one the *Jukes-Cantor*

model, which assumes a constant mutation rate for all base pairs and possibility of multiple mutations in one site. The distance derived from it can be shown to be[5]

$$d = -\frac{3}{4}\ln\left(1 - \frac{4p}{3}\right) \tag{5}$$

that for small distances ($p \sim 0$) is equal to the p-distance, because it is unlikely that multiple mutations take place in the same site.

Assume that multiple sequences of different species are known, from what has been presented so far, species i and j could be pairwise aligned and the distance d_{ij} between them could be computed with Jukes-Cantor model. If this was done for all pairs, a matrix would be created that can be used in order to build phylogenetic trees, with the so called *distance matrix methods*.

The simplest algorithm to do so is UPGMA[6], that iteratively clusters the closest two elements. When a new element C is created from C_1 and C_2, the distance matrix is updated putting[7]

$$d(C, C^*) = 0.5\, d(C_1, C^*) + 0.5\, d(C_2, C^*). \tag{6}$$

where C^* is any other element. The principal drawback of this approach is the fact that all branches have the same length, so it is only valid under the assumption of constant evolution rate for all species.

The distance method used in most situations is *neighbor-joining*[8], that clusters elements following a trade-off between minimizing the distance between them and maximizing the distances with the other elements. Once a new element is created, formula (6) is used to update the distance matrix. It has been shown that neighbor-joining produces trees that minimizes evolution under some particular circumstances[9]. This method produces unrooted trees, that can be rooted by means of the *mid-point rule*, which locates the root in the mid-point of the longest path of the tree, or with the *outgroup method* that assumes one of the species is the most distant to the others and the root is positioned to obtain a tree consistent with this hypothesis.

C. Implementation

Most scientists use Python with extra modules for bioinformatics research; however a high level language is not useful when the functions need to be programmed from scratch, because then they are significantly slower. With the use of a compiled language such as `C++` this is no longer a problem, but advanced features such as object orientation can still be used. In order to derive the results of section IV a `C++` code was developed and it has been shared under a free license in Github[10]. It can be split into three main blocks: the first is responsible for generating a pairwise alignment of two arrays of sequences using the Needleman and Wunsch algorithm and SP-scoring function, and can return the Jukes-Cantor distance between two species; the second builds phylogenetic trees with either UPGMA or neighbor-joining using a distance matrix that is automatically generated if the input is an array of all sequences; finally the third progressively aligns multiple sequences with the help of a guide tree that is created with the previous algorithms.

Specie	GB Code	Specie	GB Code
European Human	X90314	Chimp. schweinfurthii	AF176722
German neanderthal	AF011222	Chimp. troglodytes	AF176766
Russian neanderthal	AF254446	Chimp. vellerosus	AF315498
East. lowland gorilla	AF050738	Chimp. verus	AF176731
East. mountain gorilla	AF089820	Orangutan jari	AF451964
West. lowland gorilla	AY079510	Orangutan puti	AF451972

TABLE II. Names of the species used to demonstrate the algorithms reviewed in the work. The Genbank code of the original mitochondrial D-loop sequences is included for further reference; however, this sequences were modified in order to a assert that they corresponded to the same region of DNA.

IV. RESULTS AND DISCUSSION

The main algorithms presented in this work were applied to twelve sequences from species of the *Hominidae* (also known as great apes) family formed by humans, chimpanzees, gorillas and orangutans. A particular fragment of the mitochondrial D-loop was selected as the comparing region, because it has been shown that D-loop nucleotides may be useful in studies of phylogeny in vertebrates[11]. The considered species are listed in table II together with their reference code in Genbank[1].

The genetic data was used to build a guiding tree from which a multiple alignment was generated. This MSA revealed that the sequences from table II do not represent exactly the same segment of the D-loop, but some of them contained extra nucleotides at the beginning or end. This incorrect regions could be easily removed from the original sequences comparing with the multiple alignment. The whole procedure was repeated with the corrected sequences and a new MSA was generated which clearly corresponded to an alignment of analogous sites of the DNA, as can be seen in figure 1.

Once the sequences had been corrected the phylogenetic trees were rebuild with UPGMA and neighbor-joining methods, exported to *Newick format* and printed using the Biopython package. The obtained results can be seen in figure 2, and it can be observed that the topology of both trees is the same, but different to the correct one, in which the branches of the orangutans and gorillas would be swapped. It is worth noting that close species are clustered correctly and errors begin to appear when joining distant species, which is a very frequent phenomena in phylogenetics. As the topology of both trees is the same the MSA does not depend on the chosen guide tree, but it may not be the case in other situations.

The correct topology could be obtained with neighbor-joining if the outgroup method was used assuming the orangutans were the most distant species. However, with

```
Gor_Western cc-aagtattag-ctaacccatca-ataattatc-atgtatatcgtgcatcactgccagacaccatgaataatgtacggtaccataaacgc-ccaatcacctgtagcacatac-aacc-cc----cc-ccttc--c-ccccc
Gor_Eastern cctaagtattag-ttaacccacca-ataattgtc-atgtatttcgtgcattactgccagccaccatgaataatgtacggtaccataaacactcc-ctcacctataatacat---ta-c-cc----cc-cc-tca-c-ccccc
Gori_Rwanda cc-aagtattag-ttaacccacca-ataattgtc-atgtatgtcgtgcattactgccagccaccatgaataatgtacagtaccacaaacactcc-cccacctataatacat---ta-c-cc----cc-cc-tca-c-ccccc
Orangu_Jari cc--aatact-gac--c--cattt-ctaacggcctatgtatttcgtacattcctgctagccaacatgaatatcacccaacacaacaatcgcttaacc-aactataatgcatacaaaactccaa--ccacactcg-a-cctcc
Orangu_Puti cc--agtact-gac--c--cattt-ctaacggcctatgtatttcgtacattcctgctagccaacatgaatatcacccaacacaacaatcgcttaacc-aactataatgcatacaaaactccaa--ccacactcg-a-cctcc
Chimp_Trogl cctaagtatt-ggcctattcatta-c-aaccg-ctatgtatttcgtacattactgccagccaccatgaatattgtacagtactataaccactcaact-acctataatacat--taagc-ccacccccaca-ttacaacctcc
Chimp_Schew cctaagtatt-ggcttattcatta-c-aaccg-ctatgtatttcgtacattactgccagccaccatgaatattgtacagtactataatcactcaact-acctataatacat--caaac-cca-ccccaca-ttacaacctcc
Chimp_Verus cctaagtact-ggctcattcatta-c-aaccg-gtatgtacttcgtacattactgccagtcaccatgaatattgtacagtaccataatcacccaact-acctataatacat--aaaat-ccactcccaca-ccaaaaccttc
Chimp_Velle cctaagtact-ggctcattcatta-c-aaccg-ctatgtatttcgtacattactgccagccaccatgaatattgtacagtaccacaatcactcaact-acctataatacat--aaaac-ccacccccaca-tcaaaaccttc
Europ_human cc-aagtatt-gacttacccatcaac-aaccg-ctatgtatttcgtacattactgccagccaccatgaatattgcacggtaccataaatacttgacc-acctgtagtacata-aaaac-ccaat-ccaca-tcaaaaccccc
Neanthe_Rus cc-aagtatt-gactcacccatcaac-aaccg-ccatgtatttcgtacattactgccagccaccatgaatattgtacagtaccataattacttgact-acctgtaatacata-aaaac-ctaat-ccaca-tcaaccccccc
Neanthe_Ger cc-aagtatt-gactcacccatcagc-aaccg-ctatgtatctcgtacattactgttagttaccatgaatattgtacagtaccataattacttgact-acctgcagtacata-aaaac-ctaat-ccaca-tcaaaccccc
```

FIG. 1. First 160 nucleotides of the multiple sequence alignment obtained with a progressive method with SP-scoring and the UPGMA guiding tree of figure 2 (a).

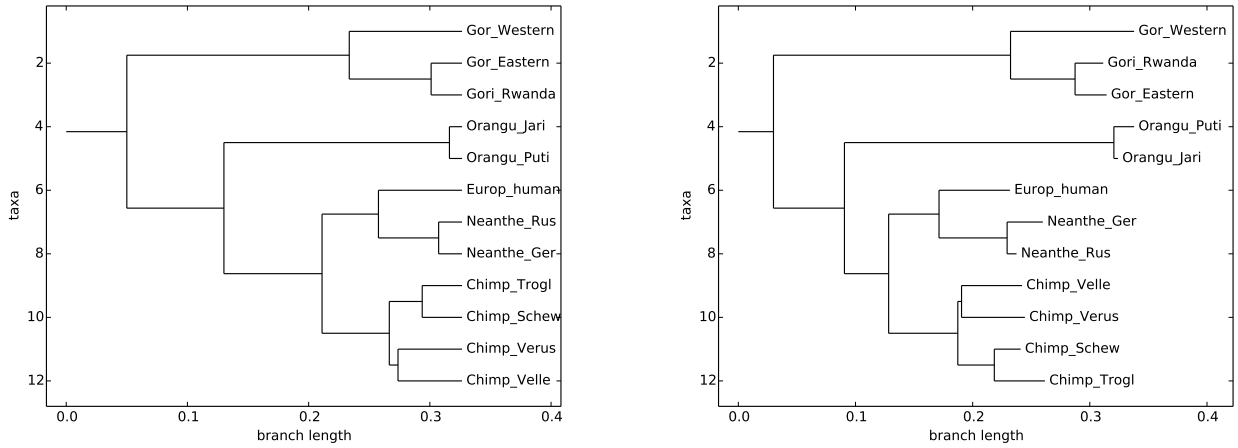

FIG. 2. A matrix d_{ij} with all pairwise distances was generated with Needleman and Wunsch algorithm with Jukes-Cantor distance, and used to generate the trees (a) with UPGMA algorithm and (b) with neighbor-joining and mid-point rooting. The genetic data is the listed in table II but with the needed modifications to assure that the same sites of the mitochondrial D-loop were compared.

the used data there was not any reasonable argument to make this assumption so this procedure was not carried out. A fast look at the MSA made it apparent that in this specific region of the D-loop the most distant species were gorillas; this indicated that a more extensive region on DNA should be probably used in order to recover the correct tree. Another possibility was that a better tree building method such as maximum parsimony, which has not been presented, could find the correct tree. However this hypothesis has not been tested in this work.

V. CONCLUSIONS

This work has shown that genetic information is a powerful tool for phylogenetic studies, and that the adequate methods can generate useful information that can be used in multiple fields of biology. The bottleneck in the development of more advanced techniques is the lack of efficient algorithms to solve the problems, so a huge field of research is open for mathematicians and computing scientists, who should find novel approaches that supersede the current ones. However, the example at the end of the paper has also shown that it is fundamental not to forget the biological expectations and that it is important to know that incorrect results may be obtained so they need to be contrasted properly.

[1] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and D. L. Wheeler, Nucleic Acids Research **33**, D34 (2005).
[2] D. W. Mount, *Bioinformatics: Sequence and Genome Analysis* (Cold Spring Harbor Laboratory Press, 2004).
[3] S. B. Needleman and C. Wunsch, Journal of Molecular Biology **48**, 443 (1970).
[4] L. Wang and T. Jiang, Journal of computational biology

1, 337 (1994).
[5] T. H. Jukes and C. R. Cantor, *Evolution of Protein Molecules*, edited by H. N. Munro (Academy Press, 1969).
[6] R. R. Sokal and C. D. Michener, University of Kansas Scientific Bulletin **28**, 1409 (1958).
[7] H. Böckenhauer and D. Bongartz, *Algorithmic Aspects of Bioinformatics*, Natural Computing Series (Springer, 2007).
[8] N. Saitou and M. Nei, Molecular Biology and Evolution **4**, 406 (1987).
[9] O. Gascuel and M. Steel, Molecular Biology and Evolution **23**, 1997 (2006).
[10] `https://github.com/aleixgg/Bioinformatics_CPP`.
[11] A. Larizza, G. Pesole, A. Reyes, E. Sbis, and C. Saccone, Journal of Molecular Evolution **54**, 145 (2002).